
Tuscany *JAVA* SCA Runtime

M2

Service Component Architecture (SCA)

- Service oriented component **model** for describing service assemblies
 - **Technology independent** modeling and composition of service networks
 - Unified, **language-independent** way to expose **implementations** as services
(Java, C++, PHP, Ruby, Javascript, BPEL, XSLT, others)
 - Binding model for **multiple access methods and infrastructure service**
 - Is not about implementing API's
 - Service components with service interfaces
-

SCA Specs

- SCA Assembly Model V0.96
 - SCA Client and Implementation Model for Java (V0.95)
 - SCA Client & Implementation for BPEL (V0.95)
 - SCA Client & Implementation for Spring (V0.95)
 - SCA Client & Implementation for C++ (V0.95)

 - SCA Web Service bindings
 - SCA JMS Binding
 - SCA EIS/JCA Bindings
 - SCA Stateless Session Bean Binding

 - SCA Policy Framework
-

Description Files within an SCA Assembly

- No deployment package specified
- ComponentType description could be a side file, annotated code or other

Assembly description

```
<?xml version="1.0" encoding="ASCII"?>
<composite xmlns=http://www.osoa.org/xmlns/sca/1.0
xmlns:foo=http://foo.com name="AccountServices">
  <service name="AccountService">
    <interface.java interface="foo.AccountService"/>
    <reference>AccountServiceComponent</reference>
  </service>
  <property name="complexFoo" type="foo:MyComplexType">
    <foo:a>AValue</foo:a>
    <foo:b>InterestingURI</foo:b>
  </property>
  <component name="AccountServiceComponent">
    <implementation.java class="foo.AccountServiceImpl"/>
    <property name="currency" source="$complexFoo/a"/>
    <reference
name="accountDataService">AccountDataServiceComponent</reference>
    <reference name="stockQuoteService">StockQuoteService</reference>
  </component>
  <component name="AccountDataServiceComponent">
    <implementation.composite name="AccountDataServiceComposite"/>
  </component>
  <reference name="StockQuoteService">
    <interface.java interface="stock .StockQuoteService"/>
  </reference>
</composite>
```

.scdl

Component Type description

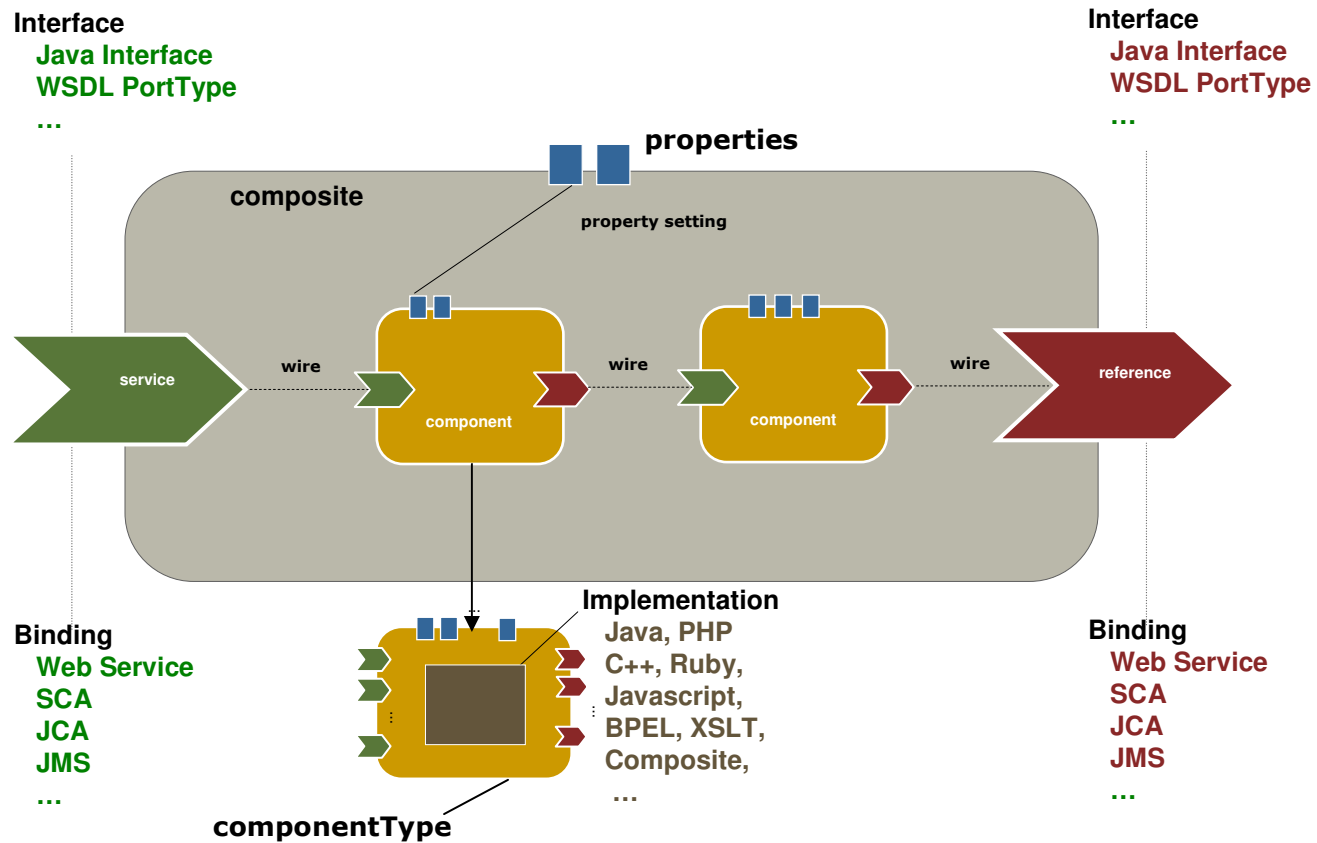
```
@Service(HelloService.class)
public class HelloServiceImpl implements HelloService {
    public String hello(String message) {
        ...
    }
}
```

Annotated code

```
<componentType xmlns="http://www.osoa.org/xmlns/sca/0.9">
  <service name="HelloService">
    <interface.java interface="services.hello.HelloService"/>
  </service>
  ...
</componentType>
```

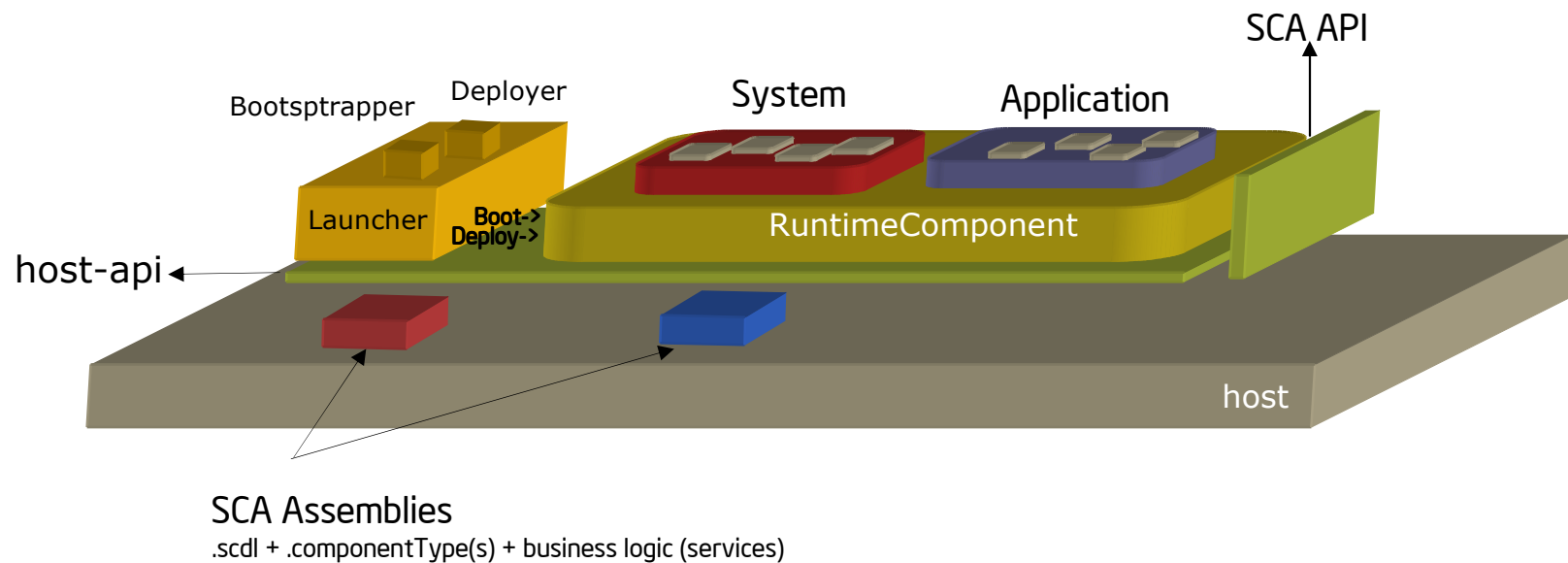
.componentType

SCA Assembly



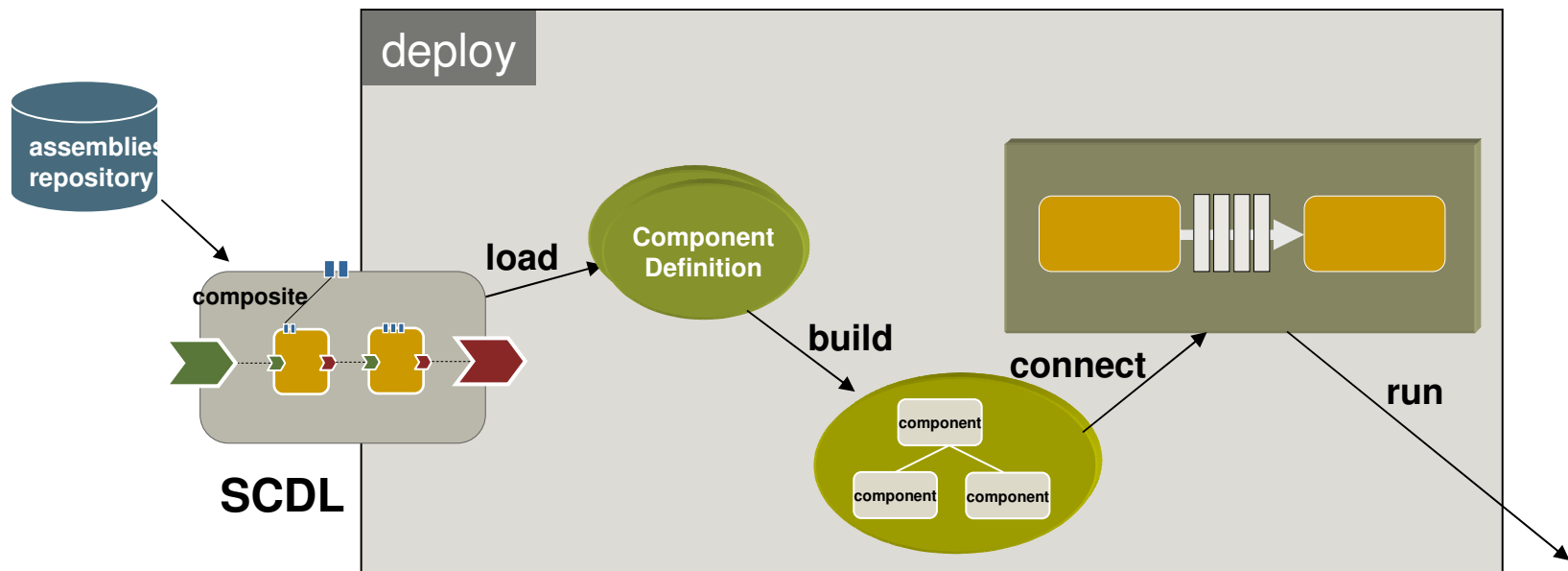
Bootstrap process

- Bootstrap process is controlled by Host environment
- Runtime is itself an SCA Assembly (“System”)



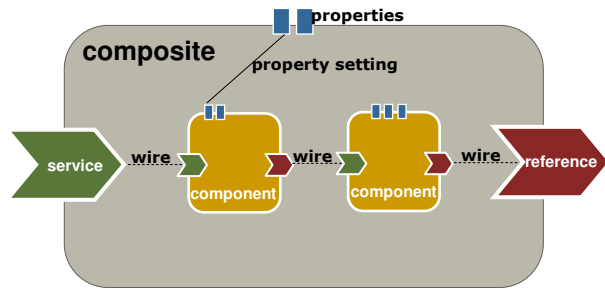
Deployment

- Deployment initiates evaluation of an assembly and wiring
 - Deploy composites or single artifacts (component, service or reference)

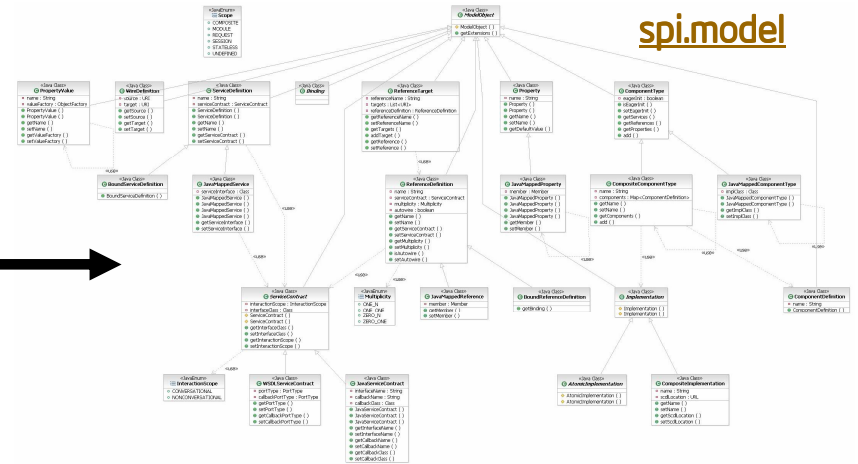


Loading Process

SCA assembly model



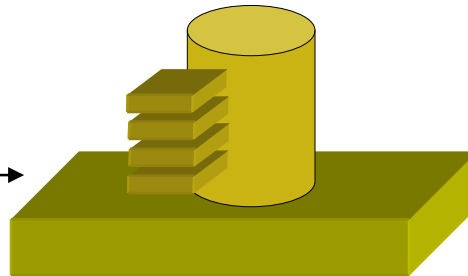
Represented by
An Object Model



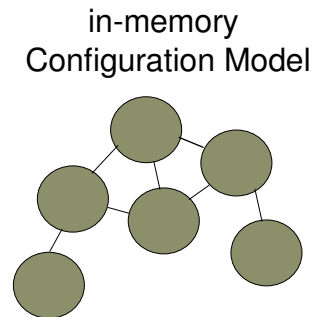
SCA assembly package
composite description (.scdl)
+ assets



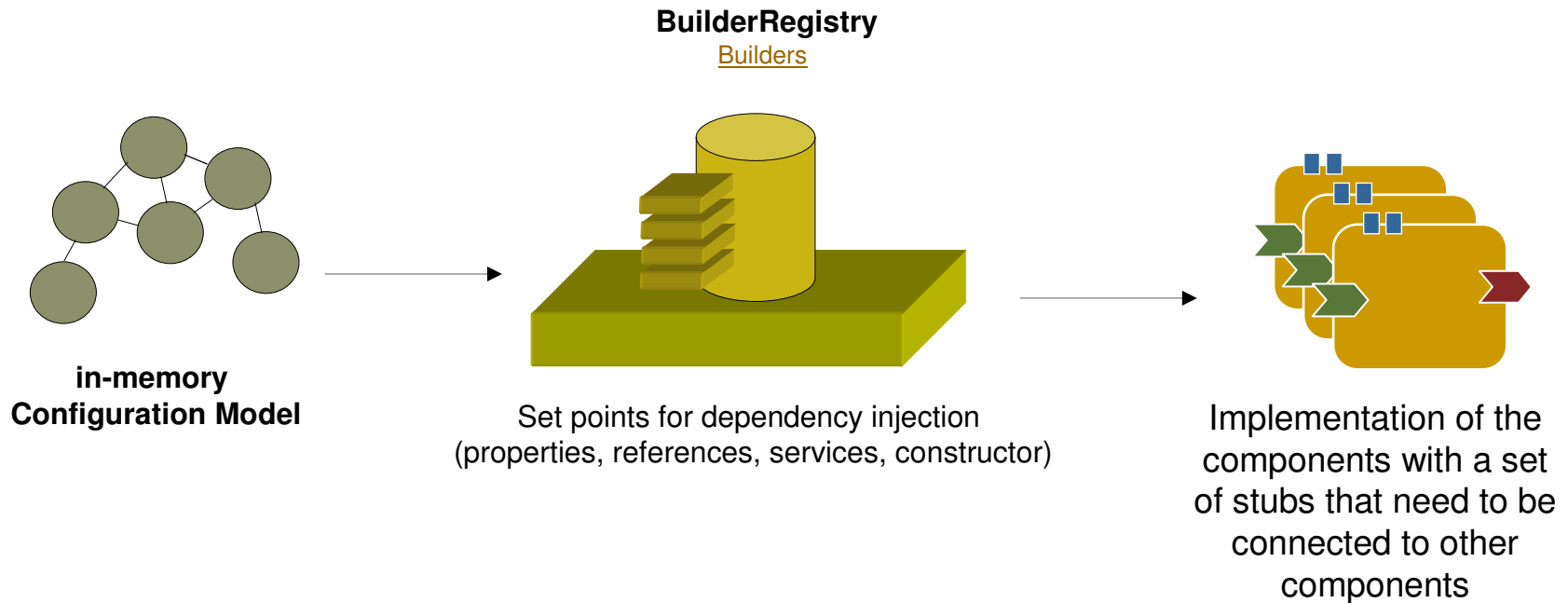
STAX/Instrospection



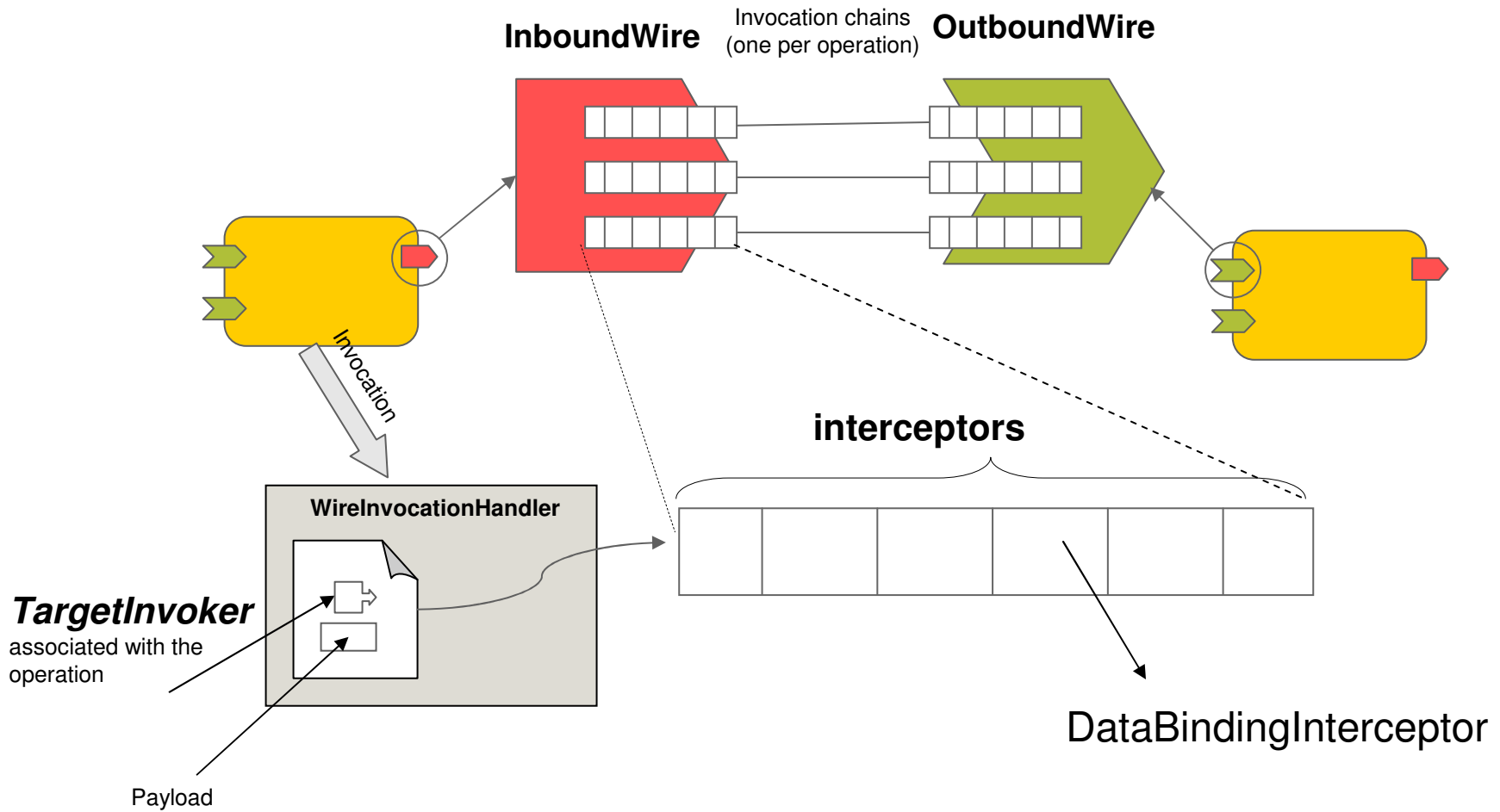
create



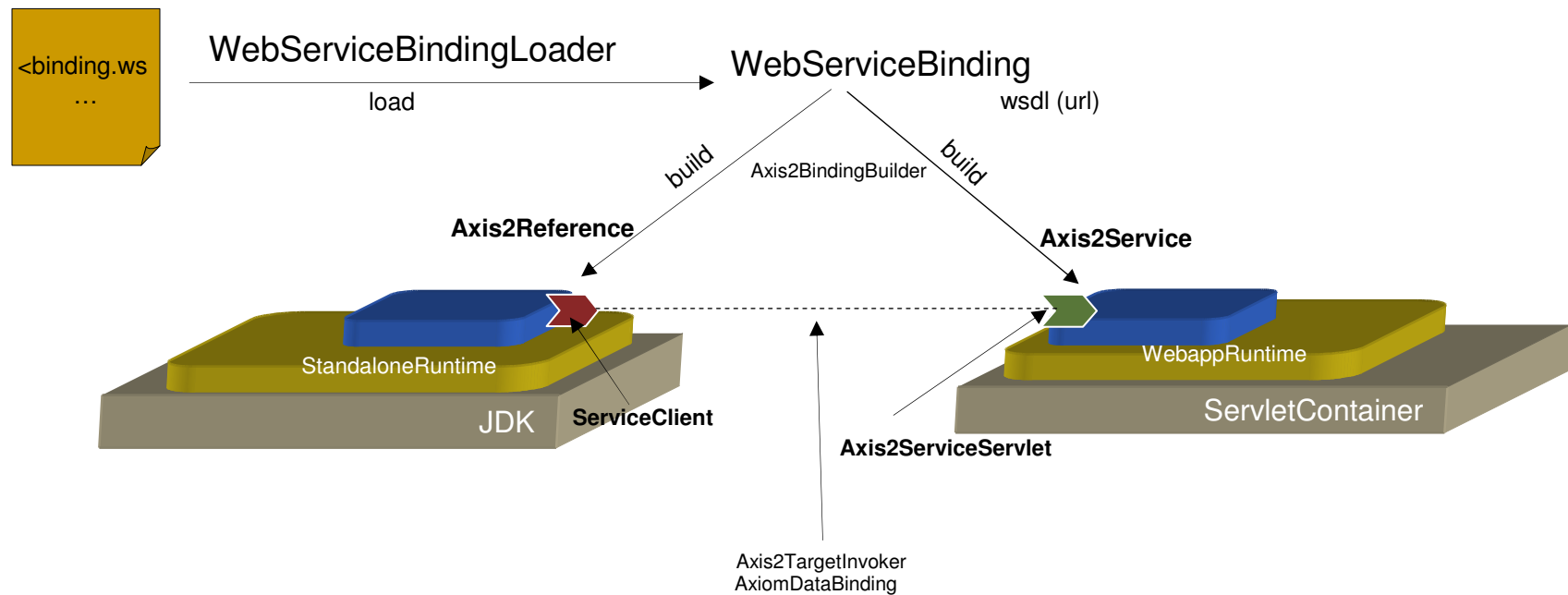
Building Process



Wiring

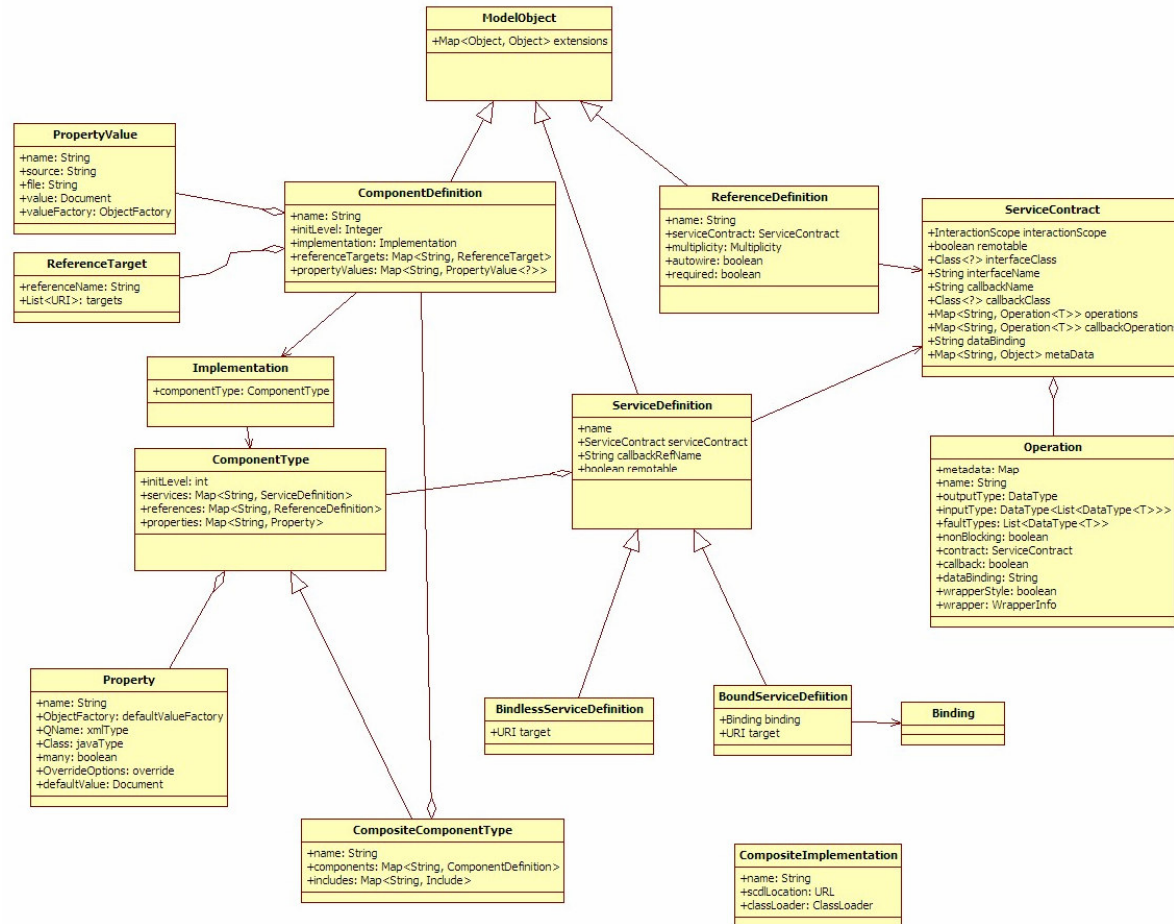


WebService Binding



Backup Info

Partial spi. model Class Diagram



Loaders

■ Core

- ComponentLoader, ComponentTypeElementLoader, PropertyLoader, ReferenceLoader, ServiceLoader, InterfaceJavaLoader, DataTypeLoader, ImplementationCompositeLoader, DependencyLoader, CompositeLoader
- **System:** SystemBindingLoader, SystemImplementationLoader, SystemComponentTypeLoader, SystemCompositeComponentTypeLoader
- **Java:** InterfaceJavaLoader, JavaComponentTypeLoader, JavaImplementationLoader

■ Extensions

- **Binding:** WebServiceBindingLoader, RMIBindingLoader, OSGiBindingLoader, JSONRPCBindingLoader
 - **Implementation:** JavaScriptImplementationLoader, RubyImplementationLoader, SpringImplementationLoader
 - **ComponentType:** JavaScriptComponentTypeLoader, RubyComponentTypeLoader
 - **IDL:** InterfaceWSDLLoader
-

Some Builders

■ Core

- CompositeBuilder, CompositeBindlessBuilder
- **System:** SystemBindingBuilder, SystemComponentBuilder
SystemCompositeBuilder
- **Java:** JavaComponentBuilder

■ Extensions

- **Binding:** Axis2BindingBuilder, RMIBindingBuilder,
ExternalJbiServiceBuilder
 - **Implementation:** JavaScriptComponentBuilder,
RubyComponentBuilder, SpringCompositeBuilder
-